CompressedStorage

Compressed Data Storage

Keeping data compressed in Hive tables has, in some cases, been known to give better performance than uncompressed storage; both in terms of disk usage and query performance.

You can import text files compressed with Gzip or Bzip2 directly into a table stored as TextFile. The compression will be detected automatically and the file will be decompressed on-the-fly during query execution. For example:

```
CREATE TABLE raw (line STRING)
   ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' LINES TERMINATED BY '\n';

LOAD DATA LOCAL INPATH '/tmp/weblogs/20090603-access.log.gz' INTO TABLE raw;
```

The table 'raw' is stored as a TextFile, which is the default storage. However, in this case Hadoop will not be able to split your file into chunks/blocks and run multiple maps in parallel. This can cause underutilization of your cluster's 'mapping' power.

The recommended practice is to insert data into another table, which is stored as a SequenceFile. A SequenceFile can be split by Hadoop and distributed across map jobs whereas a GZIP file cannot be. For example:

```
CREATE TABLE raw (line STRING)
   ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' LINES TERMINATED BY '\n';

CREATE TABLE raw_sequence (line STRING)
   STORED AS SEQUENCEFILE;

LOAD DATA LOCAL INPATH '/tmp/weblogs/20090603-access.log.gz' INTO TABLE raw;

SET hive.exec.compress.output=true;
SET io.seqfile.compression.type=BLOCK; -- NONE/RECORD/BLOCK (see below)
INSERT OVERWRITE TABLE raw_sequence SELECT * FROM raw;
```

The value for io.seqfile.compression.type determines how the compression is performed. Record compresses each value individually while BLOCK buffers up 1MB (default) before doing compression.

LZO Compression

See LZO Compression for information about using LZO with Hive.